UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/750,632	12/19/2003	Adam J. Simonoff	84734	9927

23501 7590 05/28/2008

NAVAL SURFACE WARFARE CENTER DAHLGREN DIVISION
OFFICE OF COUNSEL, CODE XDC1
17632 DAHLGREN ROAD
SUITE 121
DAHLGREN, VA 22448-5110

EXAMINER	
NGUYEN, PHILLIP H	

ART UNIT	PAPER NUMBER
2191	

MAIL DATE	DELIVERY MODE
05/28/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/750,632	SIMONOFF ET AL.	
	Examiner	Art Unit	
	Phillip H. Nguyen	2191	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *02 January 2008*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1,2,4-12,14-20,22-24,26-35,37 and 38* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1,2,4-12,14-20,22-24,26-35,37 and 38* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date *12192003*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. This action is in response to the amendment filed 1/2/2008.

2. Claims 1, 2, 4-12, 14-20, 22-24, 26-35, 37, and 38 remain pending and have been considered below.

Response to Arguments

3. Applicant's arguments with respect to claims 1 and 17 have been considered but are moot in view of the new ground(s) of rejection.

Claim Rejections - 35 USC § 101

35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

the claimed invention is directed to non-statutory subject matter.

Specifically, claims 17-24, 26-33 appear to be software per se.

Claim Rejections - 35 USC § 102

4. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

5. Claims 1, 2, 4-10, 12, 14-20, 22-24, 26-32, 34, 35, 37, and 38 are rejected under 35 U.S.C. 102(e) as being anticipated by Dye et al. (USPN: 6,802,053).

As per claim 1:

Dye teaches:

providing asynchronous access to multiple users to a graphical programming and analysis environment program visually represented as a white board (see at least FIGS. 5-6);

allowing each user to generate graphically represented code objects within the environment program (see at least col. 10:61-65 "FIG. 4 is a flowchart diagram illustrating one embodiment of how a user may interactively or manually **create or edit a graphical program**. As shown in the flowchart and described below, the user interactively **adds various objects to a graphical program**..."), *further comprising*:

allowing said each user to instantiate one or more code objects (see at least col. 10:61-65 "FIG. 4 is a flowchart diagram illustrating one embodiment of how a user may interactively or manually **create or edit a graphical program**. As shown in the flowchart and described below, the user interactively **adds various objects to a graphical program**..."),

allowing said each user to determine an internal logic for each code object (see at least FIGS. 5-6; see also at least col. 12:5-12 "output terminals of the two numeric constant nodes are connected to the input terminals of an addition function node. **The addition function node performs the addition**

operation on the numeric input. The output terminal of the addition function node is connected to the input of the user interface indicator node so that the result of the addition operation is displayed in the user interface panel"),

allowing said each user to determine first data to be received by said each code object (see at least FIGS. 5-6; see also at least col. 12:1-3 "objects may include input and output terminals and the developer may connect the output terminal of one node to the input terminal of another node"), and

allowing said each user to determine second data to be sent by said each code object (see at least FIGS. 5-6; see also at least col. 12:1-3 "objects may include input and output terminals and the developer may connect the output terminal of one node to the input terminal of another node");

allowing said each user access to the code objects of other users of the multiple users based on security privileges accorded to said each user (see at least col. 7:34-36 "**Various types of privileges or permissions may be assigned to different users**, granting them different levels of control over the graphical program");

allowing each user to have the code objects of said each user be chained to the code objects of the other users to which said each user has access to yield inter-code object communication by inter-code object connections, each inter-code object connection terminating on one of an edge and an interior of one of the code objects (see at least col. 11:65-67 "the developer may also **connect or wire the graphical program objects** in order to achieve the desired executable logic, data flow, and/or control flow"); and

allowing said each user to execute application programs made up of the code objects as chained together within the environment program (see at least col. 12:36-39 "a graphical programming environment may allow a program to be run from within the development environment, or the developer may create a standalone program and run the program, etc").

As per claim 2:

Dye further teaches:

wherein providing asynchronous access to the multiple users to the graphical programming and analysis environment program comprises enabling multiple users to log into the environment program remotely, such that the multiple users are able to access the environment program simultaneously (see at least col. 7:25-28 "multiple clients may connect to computer 86 (e.g., server 86) in order to view the graphical program's user interface panel and/or interact with the graphical program").

As per claim 4:

Dye further teaches:

wherein allowing each user access to the code objects of the other users based on security privileges accorded to the user comprises rendering visible to each user the code objects of the other users to which the user has access (see at least col. 7:34-36 "**Various types of privileges or permissions may be assigned to different users**, granting them different levels of control over the graphical program").

As per claims 5 and 37:

Dye further teaches:

wherein allowing each user to have the code objects of the user to be chained to the code objects of the other users to which the user has access comprises allowing the user to graphically chain code objects together, such that a sender object of a pair of graphically chained together code objects is able to send data that is received by a receiver object of the pair (see at least col. 11:65-67 "the developer may also **connect or wire the graphical program objects** in order to achieve the desired executable logic, data flow, and/or control flow").

As per claim 6:

Dye further teaches:

wherein allowing each user to execute the application programs made up of the code objects as chained together within the environment program comprises displaying to the user end results of data processed by the code objects upon execution of the application programs (see at least FIGS. 5-6).

As per claims 7, 18, and 38:

Dye further teaches:

wherein each code object is an applet program (see at least col. 10:8-11 "the graphical program is a program constructed using the National Instruments Lab View

graphical programming environment application, which provides specialized support for developers of instrumentation and industrial automation applications. However, it is noted that the present invention can be used for a plethora of applications and is not limited to instrumentation or industrial automation applications. In other words, FIGS. 2A and 2B are exemplary only, and users may remotely interact with graphical programs for any of various type of purposes in any of various applications").

As per claim 8:

Dye further teaches:

wherein at least one of the graphical programming and analysis environment program and the code objects is developed within an architecture- independent and Internet web browsing program-independent computer programming technology (see at least col. 10:8-11 "the graphical program is a program constructed using the National Instruments Lab View graphical programming environment application, which provides specialized support for developers of instrumentation and industrial automation applications. However, it is noted that the present invention can be used for a plethora of applications and is not limited to instrumentation or industrial automation applications. In other words, FIGS. 2A and 2B are exemplary only, and users may remotely interact with graphical programs for any of various type of purposes in any of various applications").

As per claim 9:

Dye further teaches:

wherein the graphically represented code objects coexist with. non-graphically represented code objects within the environment program (see at least col. 12:31-32 "These data structures may be compiled into machine code, or interpreted during execution").

As per claim 10:

Dye further teaches:

wherein the non-graphically represented code objects comprises stand-alone computer program (see at least col. 12:31-32 "These data structures may be compiled into machine code, or interpreted during execution").

As per claim 12:

Dye further teaches:

wherein the graphically represented code objects comprises one or more of: database objects, video-playing programs, audio-playing programs, image-viewing programs, geo-spatial information map-viewing programs, filter-algorithm programs, and model and analysis tool programs (see at least FIGS. 5-6, 8A-10B).

As per claim 14:

Dye further teaches:

wherein providing asynchronous access to the graphical programming and analysis environment program comprises providing application programs executable within the white board (see at least col. 12:36-39 "a graphical programming environment may allow a program to be run from within the development environment...").

As per claim 15:

Dye further teaches:

wherein providing the application programs executable within the white board comprises executing the application programs such that results thereof are immediately available to the multiple users (see at least FIGS. 5-6 and 8A-10B).

As per claim 16:

Dye further teaches:

wherein providing asynchronous access to the graphical programming and analysis environment program comprises allowing users to access resources available on a network to which the graphical programming and analysis environment program is communicatively coupled (see at least FIG. 7; see also col. 12:56 "Accessing a Remote Graphical Program").

As per claim 17:

Dye teaches:

a plurality of graphically represented code objects, each code object created by a user (see at least col. 10:61-65 "FIG. 4 is a flowchart diagram illustrating one embodiment of how a user may interactively or manually **create or edit a graphical program**. As shown in the flowchart and described below, the user interactively **adds various objects to a graphical program**...") *and accessible by other users in accordance with security privileges of the other users* (see at least col. 7:34-36 "**Various types of privileges or permissions may be assigned to different users**, granting them different levels of control over the graphical program"), *said each code object comprises*:

a data interface indicating first data to be input into the code object and second data to be output by the code object (see at least FIGS. 5-6; see also at least col. 12:1-3 "objects may include input and output terminals and the developer may connect the output terminal of one node to the input terminal of another node"), and

internal logic to generate the second data from the first data;

a plurality of graphically represented inter-code object connections, each inter-code object connection representing data transfer between a pair of code objects (see at least FIGS. 5-6; see also at least col. 12:5-12 "output terminals of the two numeric constant nodes are connected to the input terminals of an addition function node. **The addition function node performs the addition operation on the numeric input**. The output terminal of the addition function node is connected to the input of the user interface indicator node so that the result of the addition operation is displayed in the user interface panel");

one or more application programs composed of one or more chains of the code objects interconnected via the inter-code object connections (see at least col. 11:65-67 "the developer may also **connect or wire the graphical program objects** in order to achieve the desired executable logic, data flow, and/or control flow") and,

a graphical white board area within which the code objects are definable and movable and the inter-code object connections are creatable (see at least FIGS. 5-6); and

wherein the one or more application programs are executable within the graphical white board area, and each inter-code object connection terminates on one of an edge and an interior of one of the code objects (see at least FIGS. 5-6).

As per claim 19:

Dye further teaches:

wherein the graphical white board area is an applet program having a context within which each code object runs (see at least col. 10:8-11 "the graphical program is a program constructed using the National Instruments Lab View graphical programming environment application, which provides specialized support for developers of instrumentation and industrial automation applications. However, it is noted that the present invention can be used for a plethora of applications and is not limited to instrumentation or industrial automation applications. In other words, FIGS. 2A and 2B are exemplary only, and users may remotely interact with graphical programs for any of various type of purposes in any of various applications").

As per claim 20:

Dye further teaches:

wherein the applet program is a Java applet program (see at least col. 10:8-11 "the graphical program is a program constructed using the National Instruments Lab View graphical programming environment application, which provides specialized support for developers of instrumentation and industrial automation applications. However, it is noted that the present invention can be used for a plethora of applications and is not limited to instrumentation or industrial automation applications. In other words, FIGS. 2A and 2B are exemplary only, and users may remotely interact with graphical programs for any of various type of purposes in any of various applications").

As per claim 22:

Dye further teaches:

wherein each code object has at least one inter-code object communication graphically terminating on one of an edge and an interior of the code object (see at least FIGS. 5-6; see at least col. 11:65-67 "the developer may also **connect or wire the graphical program objects** in order to achieve the desired executable logic, data flow, and/or control flow").

As per claim 23:

Dye further teaches:

wherein each inter-code object connection represents data being sent by a sender object of the pair of code objects and being received by a receiver object of the pair of

code objects (see at least FIGS. 5-6; see also at least col. 12:5-12 "output terminals of the two numeric constant nodes are connected to the input terminals of an addition function node. **The addition function node performs the addition operation on the numeric input**. The output terminal of the addition function node is connected to the input of the user interface indicator node so that the result of the addition operation is displayed in the user interface panel").

As per claim 24:

Dye further teaches:

wherein the data are at least one of: user defined, and filtered according to security privileges accorded to the users (see at least col. 7:34-36 "**Various types of privileges or permissions may be assigned to different users**, granting them different levels of control over the graphical program").

As per claim 26:

Dye further teaches:

wherein at least one of the inter-code object connections is one of graphically invisible and purposefully limited in functionality for security (see at least FIGS. 5-6; see also 12:26-31 "The graphical program may be saved in any of various formats. For example, a tree of data structures may be built which represents the various elements of

the graphical program and the relationships among the elements, and the data structures may be saved in a binary or text format").

As per claim 27:

Dye further teaches:

wherein each inter-code object connection is graphically represented by one of a line and a directed graph (see at least FIGS. 5-6).

As per claim 28:

Dye further teaches:

wherein the one or more application programs are constructed one of asynchronously and synchronously (see at least col. 7:16 "Microsoft's Asynchronous Pluggable Protocols specification").

As per claim 29:

Dye further teaches:

wherein the one or more application programs are at least one of: capable of being stored for later retrieval and use, and modular in nature so that more complex application programs may be constructed therefrom (see at least col. 7:18-28 "Computer system 86, which may be referred to as server computer system 86, comprises a graphical program…multiple clients may connect to computer 86 in order to view the graphical program's user interface panel and/or interact with the graphical program").

As per claim 30:

Dye further teaches:

wherein the one or more application programs are contained within container panels as macro programs, the container panels inter-connectable via additional inter-code object connections (see at least FIGS. 5-6).

As per claim 31:

Dye further teaches:

wherein the one or more application programs are at least one of: auditable and loggable during usage, traceable to users who construct the programs, traceable to users who use the programs, and configuration manageable (see at least col. 7:37-42 "Various types of privilege or permissions may be assigned to different graphical program. For example, the program creator may be authorized to assume complete control over the program, locking out other user, Other users may only be authorized to view it to control the graphical program, e.g., these users may be allowed to provide input to the graphical program").

As per claim 32:

Dye further teaches:

wherein the one or more application programs are at least one of:

capable of accepting data from dynamically changing input sources, from static input sources, and from network-accessible resources (see at least col. 7:37-42 "Various types of privilege or permissions may be assigned to different graphical program. For example, the program creator may be authorized to assume complete control over the program, locking out other user, Other users may only be authorized to view it to control the graphical program, e.g., these **users may be allowed to provide input to the graphical program**"); *capable of network reporting results thereof; and, capable of networking reporting security privilege-filtered results thereof.*

As per claim 34:

Dye further teaches:

accessing by a user a graphical programming and analysis environment program that other users are already currently accessing (see at least col. 12:56 "FIG. 7 – Accessing a Remote Graphical Program");

generating by the user graphically represented code objects within the environment program, wherein for each code object (see at least col. 10:61-65 "FIG. 4 is a flowchart diagram illustrating one embodiment of how a user may interactively or manually **create or edit a graphical program**. As shown in the flowchart and described below, the user interactively **adds various objects to a graphical program**...");

the user determining a data interface indicating first data to be input into the code object and second data to be output by the code object (see at least FIGS. 5-6; see also at

least col. 12:1-3 "objects may include input and output terminals and the developer may connect the output terminal of one node to the input terminal of another node"); and,

the user determining internal logic to generate the second data from the first data (see at least FIGS. 5-6; see also at least col. 12:5-12 "output terminals of the two numeric constant nodes are connected to the input terminals of an addition function node. **The addition function node performs the addition operation on the numeric input**. The output terminal of the addition function node is connected to the input of the user interface indicator node so that the result of the addition operation is displayed in the user interface panel");

graphically chaining together code objects by the user within the environment program, including chaining together the code objects generated by the user and code objects generated by the other users to which the user has access based on security privileges accorded to the user, to yield inter-code object communication by inter-code object connections, each inter-code object connection terminating on one of an edge and an interior of one the code objects (see at least col. 11:65-67 "the developer may also **connect or wire the graphical program objects** in order to achieve the desired executable logic, data flow, and/or control flow"); and

assembling application program by the user within the environment program, each application program composed of the code objects as have been chained together (see at least col. 11:65-67 "the developer may also **connect or wire the graphical program objects** in order to achieve the desired executable logic, data flow, and/or control flow").

As per claim 35:

Dye further teaches:

executing by the user of the application programs within the environment program (see at least FIGS. 5-6).

Claim Rejections - 35 USC § 103

6. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

7. Claims 11 and 33 are rejected under 35 U.S.C. 103(a) as being unpatentable over Dye et al. (USPN: 6,802,053).

As per claim 33:

Dye does not explicitly teach:

a chat area within which the user can communicate with the other users; and, a user list area showing a name of each of the user and the other users currently logged into the environment program.

However, official notice is taken that *a chat area within which the user can communicate with the other users; and, a user list area showing a name of each of the user and the other users currently logged into the environment program* is well known to the art at the time the invention

was made for communication purposes. One would have been motivated to modify Dye's approach to include a chatting area for the users to share comments regarding the graphical program or for communication purposes.

As per claim 11:

Dye does not explicitly teach:

wherein the non-graphically represented code objects comprise one or more of: image-viewing programs, video-playing programs, and audio-playing programs.

However, official notice is taken that *wherein the non-graphically represented code objects comprise one or more of: image-viewing programs, video-playing programs, and audio-playing programs* are well known to the art at the time the invention was made. One would have been motivated to generate either image-viewing programs, video-playing programs or audio-playing programs that suits for their nature of business.

Conclusion

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Phillip H. Nguyen whose telephone number is (571) 270-1070. The examiner can normally be reached on Monday - Thursday 10:00 AM - 3:00 PM EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Wei Y. Zhen can be reached on (571) 272-3708. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

PN
4/12/2008

/Wei Zhen/
Supervisory Patent Examiner, Art Unit 2191